

# Desarrollo de un chat en el ámbito educacional sobre redes DTN.

Miguel Ángel Valverde Domínguez

**Resumen**– En el mundo actual es una realidad que existe una dependencia a estar siempre conectados a través de Internet en todo momento y en cualquier lugar, pero esto muchas veces no es posible. Ya sea por la situación geográfica o por otros motivos, no siempre se puede estar conectado a Internet, aquí es donde entran en juego las Redes Tolerantes a Cortes y Interrupciones. Con esta investigación se pretende desarrollar una aplicación web responsiva, donde se implemente un chat que trabaje sobre redes DTN cuando haya necesidad de ello. Para llegar a este objetivo se estudian varios aspectos de las DTN como son: Capacidad de actuar en un entorno tan dispar como el Espacio, algoritmos de encaminamiento, transferencia bajo custodia, entre otros.

**Palabras clave**– Tolerante, responsiva, chat, custodia, encaminamiento, Espacio.

**Abstract**– Nowadays, it's a reality that there is a dependency of being connected to Internet everywhere and at all times, but this is not always possible. Geographical position is only one reason among many. When we cannot connect to the internet we can use the Delay-Tolerant networking(DTN). The purpose of this research is to develop a responsive web application, that allows us to use a chat using the DTN technology. To achieve this purpose, we have studied various aspects like the capacity to act in an environment as strange as the Space, routing algorithms and custody transfer among others.

**Keywords**– Tolerant, responsive, chat, custody, routing, Space

## 1 INTRODUCCIÓN

EL reto propuesto en este trabajo de final de grado ha sido el de adaptar y crear un sistema de chat, pero con la peculiaridad de que este utiliza las virtudes de la tecnología DTN. Estas redes son capaces de transmitir los mensajes después de caídas en la comunicación y salvando problemas de distancias elevadas entre los nodos.

### 1.1. DTN

Debido al constante auge en el campo de la ciencia y la tecnología de las telecomunicaciones, ha aumentado la complejidad a la hora de solucionar los problemas que nos podemos encontrar a la hora de comunicarnos con y por todo el mundo. No podemos dejar de lado que no todas las zonas geográficas poseen la misma calidad, en lo que cobertura de conexión se refiere. Esto supone un verdadero problema a la hora de querer establecer una conexión con estas zonas. Las Redes Tolerantes al Retardo nos han ayudado a resolver este tipos de problemas de conexión, entre

otros, solventando así algunas carencias que presenta el sistema actual de redes. La arquitectura de estas redes añade una nueva capa (que puede ser TCP o no) entre dos capas ya existentes, la capa de redes y la capa de aplicaciones. Técnicas como la transferencia bajo custodia y el almacenamiento y reenvío le proporciona a este tipo de red una gran utilidad y favorecen su uso en zonas donde Internet y otras redes no son capaces de llegar. Aún en fase de desarrollo, las redes DTN han demostrado que pueden mejorar considerablemente las comunicaciones en zonas con dificultades comunicativas, ya sea una zona remota sin cobertura de red, un navegador GPS o incluso, como veremos más adelante, para establecer comunicaciones con algún sistema que se encuentre fuera de nuestro planeta.

Como hemos indicado anteriormente, esta red puede servir para comunicarnos fuera de nuestra planeta, de hecho, fue creada con dicho propósito. En 2002 apareció por primera vez el término DTN, cuando Kevin Fall empezó a adaptar todas las ideas surgidas en La Internet Interplanetaria(IPN) a un uso terrestre [1]. Dicho proyecto surgió con la necesidad de proporcionar una nueva forma de comunicarse entre naves espaciales, ofreciéndoles capacidades de manejar comunicaciones con importantes retrasos y interrupciones.

Las redes DTN utilizan una serie de protocolos y capas distintas de las que utiliza Internet, a priori, ya que no utiliza

- E-mail de contacto: mavalverdedominguez@gmail.com
- Mención realizada: Technologies de la Informació
- Trabajo tutorizado por: Guillermo Navarro (DEIC)
- Curso 2015/16

la pila de protocolos TCP/IP para la capa de transporte.

Las características más importantes de DTN son: su capa de fardos, la arquitectura de su red, la conectividad entre nodos y por último, la capacidad de utilizar la transferencia bajo custodia.

**Capa de Fardos** También denominada -bundle layer- [2], es una nueva capa que añaden las redes DTN. Esta capa se encuentra entre las capas de transporte y de aplicación como podemos ver en la figura 1. Por debajo encontramos la pila de protocolos que, en este caso, pueden ser comunes al del modelo TCP/IP. Esta capa es la encargada de almacenar los datos de los mensajes y, posteriormente reenviarlos a los posibles nodos de la red.

**Almacenaje y reenvío** En el momento en que hay una interrupción en la comunicación, los nodos DTN tienen la necesidad de reenviar los mensajes, para esto necesitan guardarlos en memoria física cierto tiempo antes de eliminarlos. Este tiempo de almacenamiento es mucho mayor que el tiempo que manejan los encaminadores de Internet, ya que estos solo guardan los fragmentos durante unos pocos milisegundos.

**Transferencia bajo custodia** Aunque no es necesario ni obligatorio que un nodo implemente la transferencia bajo custodia, este es un concepto importante, causante de las exitosas entregas de paquetes en las redes DTN. Aunque sea importante que los nodos implementen esta función, es paradójico que solo sea una parte opcional en la comunicación entre nodos. Su funcionamiento se basa en el almacenamiento de los mensajes en los nodos encaminadores hasta que el siguiente nodo en la ruta esté disponible.



Fig. 1: Arquitectura de una red DTN

## 1.2. Estado del Arte

En la actualidad hay varias investigaciones llevándose a cabo, con la finalidad de ofrecer una especie de Internet en el espacio. Fuera de nuestro planeta, los paradigmas que nos encontramos son diferentes a los que en la Tierra, ya tenemos solución. En el Jet Propulsion Laboratory de la NASA [3], Vincent Cerf junto a su grupo de investigación están investigando sobre la posibilidad de crear redes interplanetarias [4]. Tomando de ejemplo el protocolo TCP/IP, el cual sirve en la actualidad como protocolo de transporte en el sistema de Internet actual, la NASA pretende adaptar un modelo ya existente, pero con peculiaridades tan grandes como

la posibilidad de que hayan retrasos en la transmisión de datos, posibilidades que no se cumplen en el modelo TCP/IP actual.

Sin la aparición de las redes DTN, esta comunicación interplanetaria sería imposible. Nos enfrentamos a retos tan complejos como la necesidad de controlar parámetros derivados de la rotación planetaria y, de manejar grandes distancias entre los planetas. En la figura 2 podemos visualizar un ejemplo esquemático de una red interplanetaria.

Destacar que actualmente también se está investigando las posibilidades de esta tecnología con uso en la Tierra. Uno de los avances más recientes ha sido implementar DTN en dispositivos con sistema operativo Android [5] [6]. De esta forma se puede utilizar esta tecnología en millones de dispositivos móviles, abriendo el abanico de posibilidades hasta límites incalculables.

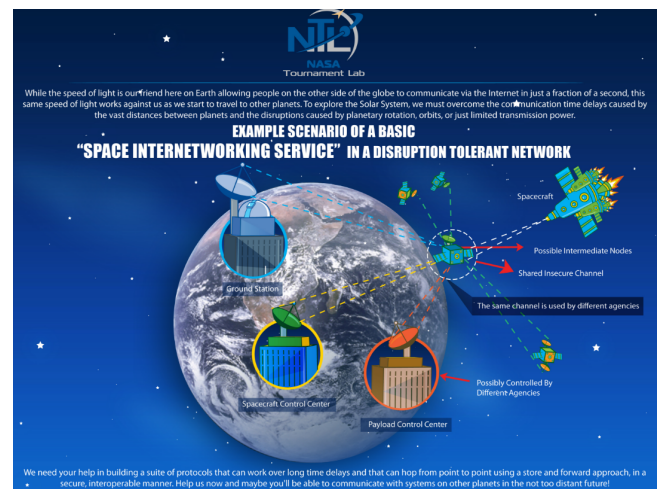


Fig. 2: Representación de una red DTN interplanetaria

## 1.3. Objetivos

Los objetivos generales que hemos intentado asumir en este proyecto han sido:

- Crear una aplicación Web capaz de representar los datos recibidos por parte del servidor, el cual procesa los datos recibidos de los nodos DTN.
- Llegar a profundizar en el tema de las Redes DTN y comprender mejor su funcionamiento y potencial.
- Desarrollar aspectos de seguridad como el cifrado de la información al ser enviada o la validación de la identidad del usuario con certificados digitales, de esta forma la red y el sistema serían del todo anónimos.
- Documentarse sobre el código inicial y re-aprovechar todo lo necesario para la nueva funcionalidad.

Una vez avanzado el proyecto y la investigación, nos dimos cuenta que hacía falta modificar los objetivos planteados inicialmente,

En esta fase del proyecto hemos podido comprobar si los objetivos iniciales pueden llegar a cumplirse y si van a ser modificados. Teniendo en cuenta que se iba a profundizar más en el diseño y la funcionalidad del portal web, que en la parte del cliente/servidor DTN.

-Se añadieron:

- Creación de un servidor HTTP que recoja información de la aplicación de chat desarrollada y de toda su gestión.  
-Se eliminaron:
- Desarrollar aspectos de seguridad como el cifrado de la información al ser enviada o la validación de la identidad del usuario con certificados digitales, de esta forma la red y el sistema serían del todo anónimos.

## 2 METODOLOGÍA

La metodología escogida para la realización de este proyecto ha sido una metodología cíclica. Primero nos planteamos una metodología “Ágil” como puede ser SCRUM o Programación Extrema (XP)[7]-[8]. La metodología SCRUM nos aportaba los beneficios de una metodología iterativa o cíclica:

- Flexibilidad a cambios.
- Mayor calidad del software.
- Mayor productividad.
- Predicciones de tiempos.
- Reducción de riesgos.

Pese a estos beneficios, ha sido inviable la ejecución de este método de forma estricta para este proyecto, ya que se ha realizado de forma individual y por lo tanto no tenía mucho sentido seguir todos los roles y las reuniones que propone SCRUM. En su defecto hemos decidido optar por una metodología más abierta que nos permitiera organizarnos mejor a mi tutor y a mi. Se ha desarrollado el código, la documentación y las ideas en función de los requisitos. El tutor ha ejercido el papel de “cliente” en el proyecto, revisando el material entregado y proponiendo posibles mejoras, lo cual ha hecho que se inicie otra vez con una iteración del desarrollo del proyecto. Las iteraciones o etapas no las hemos dado por concluidas hasta que no se han cumplido todos los requerimientos de dicha etapa. Aunque han ido cambiando los objetivos y los requisitos iniciales, esto no ha implicado ningún cambio en la metodología, todo lo contrario, esto ha hecho que podamos seguir bien la planificación del proyecto y la hayamos cumplido en la medida de lo posible sin grandes cambios. A continuación nombraré algunos de los beneficios de una metodología iterativa como la utilizada en este proyecto de carácter individual:

- El proyecto se divide en iteraciones, cuyo entregable es una versión del sistema totalmente funcional.
- Todo (planificación, análisis, diseño, código, etc.) está sujeto a ser modificado en futuras iteraciones.
- Se intenta aprovechar el aprendizaje durante el desarrollo. Esta metodología es ideal para cuando los requisitos no están claros o simplemente, pueden cambiar mucho en el tiempo.

Gracias a esta metodología, obtenemos los beneficios de una metodología iterativa y a la vez no nos vemos inmersos



Fig. 3: Ciclo de Vida de las Metodologías Ágiles

en una forma de trabajar inviable en un proyecto de ámbito individual. Cabe destacar que la metodología escogida no se ha visto prácticamente afectada en su núcleo, sí que por culpa de algunos cambios en la planificación también se ha alterado la metodología en cierta manera: en ocasiones se dejaba sin finalizar completamente una parte del trabajo para centrarse o priorizar otros aspectos más urgentes.

## 3 REQUISITOS

En un principio se planteaba la opción de hacer una aplicación de reporte de incidencias que trabajara por encima de una red DTN y por lo tanto se beneficiase de las ventajas que aporta esta tecnología. Una vez empezado el proyecto, decidimos orientar el trabajo hacia otro lugar y darle un punto de vista diferente. Decidimos hacer un chat en el ámbito educacional que trabajase por encima de HTTP y DTN. El proyecto ha ido cambiando y en principio pretendíamos abarcar más objetivos y requisitos de los que al final se han podido abarcar por falta de tiempo y por cambios en la planificación.

Al final los requisitos del sistema son:

### 3.1. Funcionales

- El sistema debe permitir la identificación y autenticación de usuarios.
- El sistema debe detectar logins erróneos y posibles intentos fraudulentos de entrar en el sistema.
- El sistema debe contener una administración y gestión.
- Poder enviar y recibir mensajes en el chat, a través de una red DTN.
- El sistema mostrará información como la fecha y el usuario que ha enviado el mensaje.
- Creación de un servidor y un cliente HTTP.
- El sistema debe contener una administración y gestión.

### 3.2. No Funcionales

- El sistema debe ser intuitivo.
- El sistema debe contener información actualizada.
- El sistema debe ser ágil y rápido.

## 4 ANÁLISIS DEL PROYECTO

En este apartado hablaremos sobre el diseño de la Base de Datos, los Casos de Uso que podemos encontrar en nuestro aplicativo y por último, habrá un apartado donde nos acercaremos a las tecnologías más importantes utilizadas en este proyecto.

### 4.1. Base de Datos

A continuación entraremos a definir nuestra Base de datos, el diagrama lógico lo podemos encontrar en la figura 4.

**Mensajes** La tabla Mensajes contiene toda la información de la parte de Chat de la aplicación. Almacena el identificador del mensaje, la fecha de envío del escrito, el remitente del mensaje, la sala desde donde se envía y se puede visualizar dicho mensaje y, por último, el texto o contenido del mensaje en sí. El campo Sala.Id es una Clave Ajena de la tabla Salas, la cual contiene el Identificador de la sala en que está el mensaje presente.

**Usuarios** La tabla Usuarios contiene toda la información referida a los usuarios, sea cual sea su rango en cuestión. Almacena el identificador del usuario, el nombre dentro del sistema, la fecha de creación dentro del aplicativo y como último campo importante, el rol o rango dentro del sistema.

**Salas** La tabla Salas contiene la mayoría de información referida a las Salas de Chat. Almacena el identificador de la Sala, el nombre visible de la Sala, la fecha de creación, etc.

**Roles** La tabla Roles contiene la información de los roles de los usuarios. Podemos encontrar tres roles, ya que estos son los posibles de un usuario registrado en la base de datos. Los roles posibles, a priori son:

- Administrador: En la actualidad solo se dispone de un administrador, con acceso a todo el contenido de la base de datos.
- Alumno: Son todos los usuarios añadidos por los profesores.
- Profesor: Tienen el mayor rango y con más permisos, después del administrador.

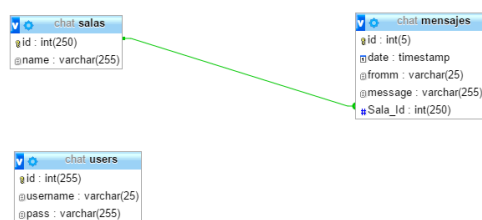


Fig. 4: Diagrama Lógico de la Base de Datos

### 4.2. Casos de Uso

Esta arquitectura de sistema se puede aplicar a muchos sistemas distintos con acceso restringido. Los casos de uso que mostramos a continuación los podemos ver gráficamente en un diagrama de casos de uso en el anexo A.2. Cada perfil va sumando las permisos o opciones de los perfiles anteriores, quitando alguna excepción.

#### 4.2.1. Invitado

Los usuarios que acceden como invitados pueden hacer login e identificarse.

#### 4.2.2. Alumno

Un usuario —previamente autenticado como alumno válido— puede acceder a la sala de mensajes en las que tenga permiso para acceder.

#### 4.2.3. Profesor

Un usuario con rango de Profesor puede crear, modificar, eliminar salas de chat y añadirle alumnos a dicha sala. Puede gestionar todos los alumnos que se encuentren dentro de su alcance.

#### 4.2.4. Administrador

Un usuario con rol de administrador puede visualizar todo el contenido (chats, usuarios de todos los rangos, mensajes) de la web. También puede crear, eliminar y modificar salas, usuarios y asignarles rangos a cada uno.

### 4.3. Tecnologías Utilizadas

#### 4.3.1. PHP:PDO

PHP(Hypertext Preprocessor) es un lenguaje de código abierto utilizado para el desarrollo web y que generalmente es utilizado para ser incrustado en HTML. PDO (Objetos de Datos de PHP) define una interfaz para poder acceder a las bases de datos en PHP. PDO nos aporta una capa de abstracción a nivel de acceso a los datos, la cual nos permite programar una vez el código de nuestra base de datos sin importar el servidor SQL que utilizemos.

#### 4.3.2. JavaScript: JQuery

Es un Frameworks de código abierto que, ofrece una serie de funcionalidades basadas en JavaScript, que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

### 4.3.3. Bootstrap

Bootstrap, es un framework CSS y JavaScript creado por Twitter, que facilita y ayuda en la maquetación de cualquier portal web. Gracias a su extensa librería y a sus archivos CSS permite el desarrollo de portales web sin la necesidad de comprender a fondo CSS. Una de las principales ventajas que nos ofrece es la de adaptar la interfaz y el diseño del portal web al tamaño del dispositivo que la visualice en ese momento. Además, Bootstrap nos aporta una creación de diseños simple, intuitivos y limpios. Esta característica aporta rapidez a la hora de renderizar la web y adaptarse a diversos dispositivos. Dicha simplicidad viene dada gracias a varios elementos predefinidos dentro del diseño de bootstrap, como pueden ser: botones, menús desplegables y/o formularios avanzados.

### 4.3.4. SeNDA

La librería SeNDA está desarrollada por el departamento del DEIC dentro de la Escuela de Ingeniería de la UAB. Una de sus funcionalidades es la de ayudarte a crear una red de nodos DTN y hacer de guía a la hora de tratar los bundles.

### 4.3.5. Telnet

Telnet, es el acrónimo de Telecommunication Network. Es un protocolo de red que normalmente se utiliza para acceder a un ordenador y manipularlo de forma remota. En esta ocasión hemos creado un cliente Telnet a través de la conexión socket. Hemos utilizado esta tecnología para enviar peticiones HTTP al servidor [9].

## 5 DESARROLLO

En este apartado reflejaremos y explicaremos como se ha llevado a cabo el desarrollo de nuestro proyecto. Dividiremos este apartado en las secciones más importantes de este.

### 5.1. Plataforma SeNDA Y DTN

Este apartado ha sido uno en los que mas tiempo se ha invertido. Lo primero ha sido realizar un estudio exhaustivo de la arquitectura de las redes DTN. Una vez entendidas estas tuvimos que dedicar muchos recursos –tiempo de aprendizaje– en el estudio de la plataforma de comunicación ADTN, desarrollado por SeNDA [10]. Esta plataforma es una red Tolerante a Interrupción y Retardo, siguiendo los principios que permite ActiveDTN, nos permite crear y enviar bundles o fardos, utilizando DTN. Gracias a la existencia de esta plataforma o librería, se ha podido aprovechar mucho trabajo y recursos para este proyecto, en definitiva código que se ha utilizado para la creación de la red de nodos DTN.

A la hora de crear la red de nodos DTN, como se ha comentado anteriormente, utilizamos la librería aDTN, la cual pudimos instalar de una forma no demasiado complicada, gracias a las instrucciones que ofrecía la documentación de la propia librería [11]. Creamos una red de varios nodos con protocolo DTN, siempre sobre Linux [12], el sistema operativo que nos permite la librería. El esquema resultante es

parecido al esbozo que encontramos en el anexo aped.A sobre una red DTN, lo que se encuentra dentro de la nube, serían los nodos DTN.

Cabe destacar, como se indicará más adelante en la parte de resultados, que el objetivo de hacer funcional la red de nodos no ha podido cumplirse, ya que ha faltado implementar parte del cliente y del servidor DTN.

### 5.2. Instalación y Configuración del cliente y la Web App

```
def recvThread(configPath, configPort):
    """Receiving thread.

    This functions opens an aDTN socket, and listens with it.
    Every new incident received is processed.
    """
    sock = adtn.adtnSocket(configPath)
    sock.bind(configPort)
    while not end:
        received = sock.recv()
        thread.start_new_thread(incidentWorker, (received,))
```

Fig. 5: Código del Incident Manager

Siguiendo la documentación que nos ofrecía el módulo de incidencias [13] del que originalmente parte este proyecto, se consiguieron instalar los requisitos necesarios para la puesta a punto de dicho Incident Manager y la App Web en un ordenador personal con sistema operativo Linux. Un ejemplo del código del que hablamos lo encontramos en la figura 5, donde observamos una función que recibe un Thread, abre un socket DTN y se mantiene a la espera de recibir nuevas incidencias para procesarlas.

Por razones de falta de entendimiento del código y sobre todo por la planificación, no se pudo completar el aprovechamiento de la mayoría de funcionalidades y, por lo tanto, se decidió junto con el tutor dejar todo preparado para una futura implementación del cliente y del servidor DTN.

#### 5.2.1. Separación Del Código Del Cliente

Para esto se utilizó Telnet junto a PHP, con el fin de ahorrar tiempo en el desarrollo de un cliente HTTP separado. Se ha separado esta parte en diferentes apartados, en los cuales se explicará el desarrollo de la web del chat, el servidor y el cliente, por separado.

#### 5.2.2. Desarrollo De La Plataforma Web Y Del Servidor HTTP

Para el desarrollo del portal web y del servidor HTTP utilizamos las herramientas que nos trae el paquete de herramientas web XAMPP [14], aprovechando el servidor Apache [15] que trae instalado y el gestor de base de datos MariaDB [16], en su versión 10.0.24. Esta implementación de base de datos deriva de MySQL [17], añadiendo nuevos motores de almacenamiento. La interfaz gráfica utilizada para el manejo de la base de datos ha sido PHPMyAdmin [18].

La web cumple con el patrón de diseño y patrón arquitectónico MVC [19], gracias al cual hemos podido separar



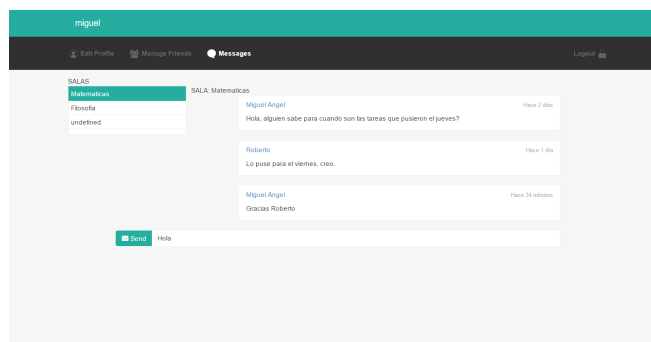


Fig. 6: Vista del Chat

la lógica de negocio de los datos. Dicho patrón nos ha ayudado a organizar el código en diferentes archivos y a mantener una lógica en la estructura de la aplicación.

### 5.2.3. Servidor HTTP

Aunque haya una separación de código para el envío de mensajes dependiendo de si se realiza por HTTP o por DTN, hay una parte importante de gestión de la aplicación que se sigue manteniendo conjunta en el servidor y el cliente Apache. Esta funcionalidad ha sido diseñada e implementada utilizando los lenguajes de programación PHP, HTML, JavaScript y CSS con Bootstrap [20] y manteniendo una coherencia en el diseño del código orientado a objetos, tratando de utilizar patrones de diseño en la medida de lo posible. En la figura 6 podemos observar un ejemplo de la vista principal del chat y, en la figura 7 la del manejo y gestión de los usuarios del sistema. Estas vistas cambiarán dinámicamente dependiendo del grado de permisos que tengamos dentro del sistema.

De la vista de chat podemos observar que simplemente con seleccionar la sala, esta cambia

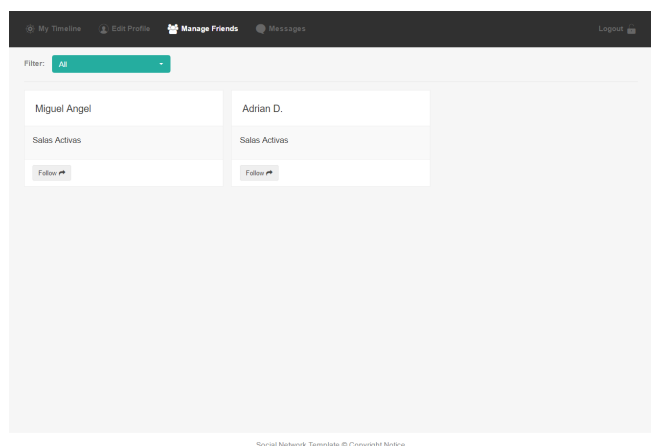


Fig. 7: Vista de la Gestión de Usuarios

### 5.2.4. Front-End Responsivo

Hemos implementado un diseño responsivo a través de CSS, gracias a las CSS Media Queries [21]. Estas nos permiten hacer un diseño que adapta el estilo de la web a cualquier resolución que hayamos diseñado. Primero se empezó especificando las funciones multimedia, las cuales hemos utilizado para definir aspectos como: color, tamaño mínimo

y máximo de alguna resolución específica, orientación de la pantalla, etc. Por último decidimos añadir las extensiones CSS de Mozilla [22].

### 5.2.5. Creación Y Utilización Del Cliente HTTP

Como hemos indicado anteriormente, decidimos separar el servidor del cliente HTTP, así obtenemos un mayor nivel de abstracción y podríamos en un futuro finalizar y reaprovechar el código del cliente y la red DTN. En la figura 8 se puede observar nuestro cliente HTTP, con el que se pueden enviar peticiones HTTP a un dominio y visualizar la respuesta obtenida del propio servidor. En nuestro cliente se pueden especificar parámetros básicos y necesarios como el dominio y puerto y se pueden agregar otros parámetros más avanzados como el timeout, el método de envío o la ruta relativa. Por último, se puede agregar cualquier cabecera HTTP al mensaje (keep-alive, etc).

#### CLIENTE HTTP

No es necesario incluir Connection: Close al final de la cabecera, pues será agregado en el script.

Dominio:   
 Protocolo:  Método:  Puerto:  Timeout:   
 Ruta relativa:   
 Cabeceras:   
 Contenido:

Fig. 8: Vista de una respuesta del servidor HTTP

Como podemos observar en la figura 9, la respuesta que nos devuelve el servidor es un código HTML que coincidirá con la petición que le hemos hecho. También cabe destacar que, una vez realizada la petición, el código de nuestro servidor ha recibido bien la petición y nos ha introducido en la base de datos los datos de usuario, mensaje y sala que le hemos indicado, en sus respectivas tablas. Esto ha sido posible gracias a que en la petición también le hemos pasado la sesión correspondiente. Si no hubiese sido así, nuestro servidor nos hubiese devuelto un mensaje en el cual nos indicaría que el usuario no está dado de alta en el sistema o, en su defecto, que no ha hecho el ingreso correspondiente.

## 6 RESULTADOS

Cabe destacar primero los resultados obtenidos del aprendizaje de la tecnología DTN, conocimiento adquirido prácticamente de cero, ya que anteriormente a este proyecto lo único que conocíamos era su funcionamiento en general, sin entrar en detalles como su capa de fardos o la transferencia bajo custodia. Otro resultado que me gustaría reflejar es el del aprendizaje a la organización individual, el cual hemos desarrollado a través de la metodología y la experiencia personal.

En el siguiente apartado comentaremos los resultados que son más perceptibles desde fuera.

Como objetivos principales del proyecto teníamos:

## Respuesta desde el servidor

```

HTTP/1.1 200 OK
Date: Mon, 30 May 2016 03:43:39 GMT
Server: Apache/2.4.17 (Ubuntu) OpenSSL/1.0.2d PHP/5.6.19
X-Powered-By: PHP/5.6.19
Content-Length: 5288
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="">
  <title>Chat Sólere DTN</title>

  <link href=".../css/default.css" rel="stylesheet">
</head>
<body>

  <!-- Fixed navbar -->
  <div class="navbar navbar-main navbar-primary navbar-fixed-top" role="navigation">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#main-nav">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
      <div data-toggle="sidebar-chat" class="btn btn-link navbar-btn visible-xs"><i class="fa fa-comments"></i></div>
    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse id="main-nav">
      <ul class="nav navbar-nav">

        <ul class="nav navbar-nav navbar-right">

          <!-- User -->
          <li class="dropdown">
            <a href="#" id="username" class="dropdown-toggle user" data-toggle="dropdown">
              <span class="caret"></span>
            </a>
            <ul class="dropdown-menu" role="menu">
              <li><a href="user-private-profile.html">Profile</a>
              </li>
            </ul>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

Fig. 9: Vista de una petición HTTP al servidor

- Crear una aplicación Web capaz de representar los datos recibidos por parte del servidor que procesa los datos recibidos de los nodos DTN.
- Llegar a profundizar en el tema de las Redes DTN y comprender mejor su funcionamiento y potencial.
- Documentarse sobre el código inicial y re-aprovechar todo lo necesario para la nueva funcionalidad.
- Creación de un servidor HTTP que recoja información de la aplicación de chat desarrollada y de toda su gestión.

Y una vez finalizado el proyecto hemos conseguido obtener los siguientes resultados:

- Aplicación web responsiva.
- Código PHP del servidor HTTP.
- Código PHP del cliente HTTP.
- Base de datos en MariaDB.
- Código HTML, JavaScript, CSS, y PHP para el portal web.

Como explicamos más adelante, no se han conseguido realizar todos los objetivos planteados al inicio del proyecto. En contraposición, también debemos comentar que se han realizado nuevas funcionalidades en la web como es una gestión más elaborada de la aplicación, y de un diseño web responsivo y más elaborado del pensado inicialmente.

## 7 CONCLUSIONES

De la realización de este proyecto podemos extraer las siguientes conclusiones:

- El grado de satisfacción relevante al proyecto es satisfactorio, ya que se han podido cumplir la mayoría de requisitos y objetivos iniciales. La planificación inicial era exigente y debido a problemas externos el trabajo

no se pudo empezar a realizar hasta tres o cuatro semanas más tarde de la fecha estipulada. Aunque no todos los objetivos pensados inicialmente se han podido desarrollar en su totalidad (sobre todo los apartados referentes a DTN), la aplicación cumple el objetivo principal de la creación de un chat funcional, la gestión completa de la sala del chat, de los usuarios y de la base de datos. Aunque no se haya llegado al cien por cien de lo que se deseaba, estamos satisfechos con los resultados obtenidos una vez finalizado el proyecto.

- Cabe destacar como hemos evolucionado en estos cuatro meses, aprendiendo y entrando en profundidad en esta nueva tecnología de comunicación. Adquiriendo los conocimientos necesarios para poder hacer un cliente HTTP desde cero, una web con diseño moderno y responsiva y, profundizando en el conocimiento tanto de lenguajes de programación como en la forma de planificarse en un proyecto individual.

## 7.1. Problemas Surgidos

Uno de los motivos que ha impedido el total desarrollo de los objetivos planteados ha sido por problemas de entendimiento con la propia librería SeNDA y, en ocasiones con el código inicial que se nos había facilitado. Del código inicial destacar, como se ha dicho anteriormente, que se trataba de un ejemplo de cliente desarrollado en Python [23], en el cual se implementaban las funciones necesarias para la lectura de la base de datos y la recepción de los mensajes DTN. A priori, no parecía una tarea complicada entender el código y poder aplicarlo, pero en el momento de empezar a desarrollar nuestro propio servidor DTN, nos dimos cuenta de lo contrario.

## 7.2. Posibles Mejoras

### 7.2.1. Mejoras en el Portal Web

A continuación comentaré algunas de las posibles mejoras en la web:

- Implementación de un histórico de los mensajes, el cual se recargase automáticamente cuando se envía o se recibe, ya que en la actualidad, como he comentado anteriormente, es necesario apretar un botón de recarga de la lista de mensajes. Esta parte no se ha mejorado por falta de tiempo en la planificación, pero la idea de como llevarlo a cabo ya estaba concebida. Se debería notificar a todos los usuarios de que ha habido un cambio en el chat de la sala en cuestión, para ello podríamos añadir un campo al mensaje y si se ha recibido dicho campo, actualizar el chat correspondiente.
- Extensión de la parte de gestión, donde se visualice toda la información de una posible red de nodos DTN. Creemos interesante para este proyecto que el administrador de la web, o un usuario con perfil más técnico tuviese acceso a una sección de la web, o en su defecto a otro portal desarrollado con el fin anteriormente mencionado.

### 7.2.2. Mejoras en el Cliente HTTP

Posibles mejoras en este apartado se verían representadas por la ampliación de las posibilidades en la misma interfaz gráfica. Aunque se pueda hacer cualquier tipo de consulta HTTP con nuestro cliente, sí que sería conveniente ampliar el servicio ofrecido, pudiendo escoger más opciones de cabeceras o de opciones, por ejemplo.

## 7.3. Líneas de Continuación

En lo primero que se debería pensar si se pretendiera la continuación de este proyecto, sería el de acabar el objetivo que no ha llegado a cumplirse, el cual nos proponía el reto de hacer toda la comunicación entre nodos y entre los propios clientes y el servidor utilizando los protocolos tolerante a retardos. La infraestructura actual ya tiene en cuenta la posible integración del protocolo DTN, además de toda la red de nodos tolerantes al retardo y interrupciones.

## AGRADECIMIENTOS

Quisiera agradecer primero a mi familia más cercana, empezando por mi hermano y mis padres, que han sabido darme fuerzas cuando más lo he necesitado, durante el desarrollo de este trabajo y durante los cuatro años de grado. Segundo agradecer a mi tutor Guillermo Navarro la dedicación y paciencia que ha demostrado tener durante el desarrollo de este proyecto. Por último agradecer a compañeros que he tenido durante la carrera universitaria, sin todos ellos no sería una realidad todo esto.

## REFERENCIAS

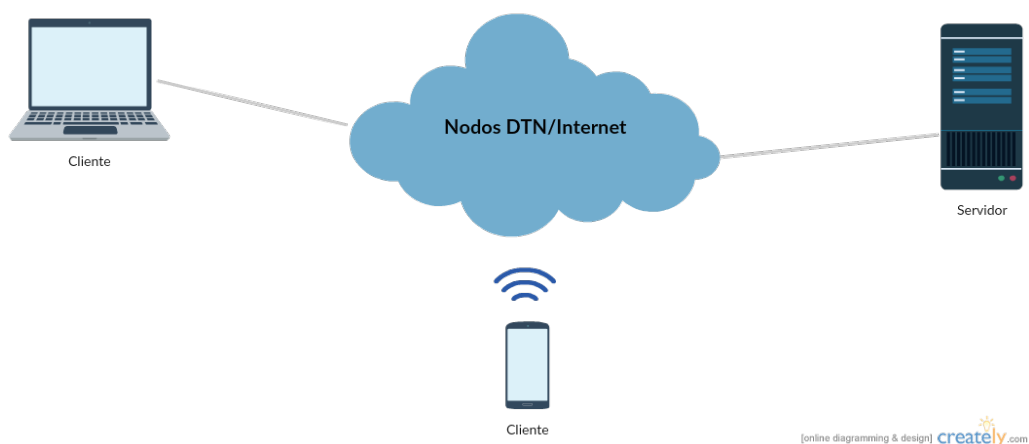
- [1] wikipedia.org, 'Delay-tolerant networking', 2016. [Online]. Available: [https://en.wikipedia.org/wiki/Delay-tolerant\\_networking](https://en.wikipedia.org/wiki/Delay-tolerant_networking) [Accessed: 15-May-2016].
- [2] L. Wood, W.M. Eddy, y P. Holliday. 'A bundle of problems', 2009. [Book] [Accessed: 09-Abr-2016] pages: 1-17.
- [3] Rhea Borja, Jet Propulsion Laboratory, 'NASA Tests First Deep-Space Internet', 2008 <http://www.jpl.nasa.gov/news/news.php?release=2008-216>.
- [4] NASA, International Space Station, 'Disruption Tolerant Networking for Space Operations (DTN)', 2016. [Online]. Available: [http://www.nasa.gov/mission\\_pages/station/research/experiments/730.html](http://www.nasa.gov/mission_pages/station/research/experiments/730.html). [Accessed: 09-Mar-2016].
- [5] Android, 'La historia de Android', 2016. [Online]. Available: <http://docplayer.es/1898855-Metodologias-iterativas-de-desarrollo.html>. [Accessed: 01- Mar-2016]. <https://www.android.com/intl/es-es/history>.
- [6] <https://github.com/h0gar/Bytewalla>.
- [7] Carlos Leone, Universidad Tecnológica Nacional, 'Metodologías Iterativas de Desarrollo', 2005. [Online]. Available: <http://docplayer.es/1898855-Metodologias-iterativas-de-desarrollo.html>. [Accessed: 01- Mar-2016].
- [8] Schenone Marcelo Hernán, UBA, 'Diseño de una Metodología Ágil de Desarrollo de Software', 2004. [Book]. Available: <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>. [Accessed: 09- Mar- 2016].
- [9] nuphsphere.com, 'HttpRequest::send', 2016. [Online]. Available: <http://www.nusphere.com/kb/phpmanual/function.http-request-send.html> [Accessed: 15- May- 2016].
- [10] SeNDA, 'Active-DTN — SeNDA', 2016. [Online]. Available: <http://senda.uab.es/Active-DTN>. [Accessed: 01- Apr- 2016].
- [11] SeNDA, 'aDTN-Platform', 2016. [Online]. Available: <https://github.com/SeNDA-UAB/aDTN-platform> [Accessed: 04- Mar- 2016].
- [12] Linux para Hispanohablantes, 'Sobre Linux', 2015. [Online]. Available: [http://www.linux-es.org/sobre\\_linux](http://www.linux-es.org/sobre_linux) [Accessed: 02-Jun-2016].
- [13] SeNDA, 'Event Application — Incident Manager', 2016. [Online]. Available: <http://tao.uab.cat/gitweb/?p=event-application.git;a=tree;f=incident-manager;h=3a1031f90d7a9f08a51fc3d69a7b07f05f38a544;hb=HEAD>. [Accessed: 04- Mar- 2016].
- [14] ApacheFriends, '¿Qué es XAMPP', 2016. [Online]. Available: <https://www.apachefriends.org/es> [Accessed: 04- Mar- 2016].
- [15] Apache, Apache Software Foundation, 2016. [Online]. Available: <https://httpd.apache.org/> [Accessed: 16-Junio-2016].
- [16] MariaDB, MariaDB Foundation, 2016. [Online]. Available: <https://mariadb.org/> [Accessed: 01-Mar-2016].
- [17] Carlos Leone, Universidad Tecnológica Nacional, 'Metodologías Iterativas de Desarrollo', 2005. [Online]. Available: <https://www.mysql.com/> [Accessed: 01- Mar- 2016].
- [18] PHPMyAdmin, 'About', 2016. [Online]. Available: <https://www.phpmyadmin.net/> [Accessed: 05- Feb- 2016].
- [19] LibrosWeb, 'La arquitectura MVC', 2016. [Online]. Available: [http://librosweb.es/libro/jobeeet\\_1\\_4/capitulo\\_4/la\\_arquitectura\\_mvc.html](http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html) [Accessed: 08- Mar- 2016].
- [20] Alvaro Fontela, '¿Qué es Bootstrap', 2015. [Online]. Available: <https://raiolanetworks.es/blog/que-es-bootstrap/> [Accessed: 04- Mar- 2016].



- [21] W3schools, 'Responsive Web Design - Media Queries', 2016. [Online]. Available: [http://www.w3schools.com/css/css\\_rwd\\_mediaqueries.asp](http://www.w3schools.com/css/css_rwd_mediaqueries.asp) [Accessed: 01-Jun-2016].
- [22] MDN, 'Extensiones CSS de Mozilla', 2016. [Online]. Available: [https://developer.mozilla.org/es/docs/Web/CSS/Referencia\\_CSS/Extensiones\\_CSS\\_Mozilla](https://developer.mozilla.org/es/docs/Web/CSS/Referencia_CSS/Extensiones_CSS_Mozilla) [Accessed: 11-Jun-2016].
- [23] Python, 'About Python', 2016. [Online]. Available: <https://www.python.org/> [Accessed: 26-May-2016].

## A ANEXO

### A.1. Esquema de Red



### A.2. Diagrama de Casos de Uso de la Web

